Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Please cancel claims 1-5, 9-12, and 16 without prejudice. Please amend claims 6 and 13. Please add new claims 17-22.

Claims 1-5 (canceled)

6. (currently amended): A ~~second~~ type very long instruction word (VLIW) memory (VIM) direct memory access (DMA) apparatus comprising:

a partitioned VLIW memory (VIM) having a separate VIM section per VLIW slot function unit;

a DMA interface;

a DMA very long instruction word (VLIW) line buffer; and

a VIM load controller for separately controlling the loading of each separate VIM section.

7. (original): The apparatus of claim 6 wherein each separate VIM section has two ports allowing simultaneous read and write accesses.

8. (original): The apparatus of claim 6 wherein said line buffer receives and temporarily stores a data packet comprising a load/modify VLIW memory address (LV2) instruction and a plurality of short instruction words (SIWs) constituting a specified functional VIM portion to be loaded at an address specified in the LV2 instruction.

Claims 9-12(canceled)

13. (currently amended): A ~~second~~ type of method for providing very long instruction word (VLIW) memory (VIM) direct memory access (DMA), said method comprising the steps of:

storing a DMA very long instruction word (VLIW) line buffer;

utilizing a VIM load controller for separately controlling the loading of each separate VIM section in a partitioned VLIW memory (VIM) having a separate VIM section per VLIW slot function unit; and

selectively routing the appropriate portions of said VLIW from said line buffer to said separate VIM sections on a DMA interface.

14. (original): The method of claim 13 wherein each separate VIM section has two ports allowing simultaneous read and write accesses, and the method further comprises the step of separately providing each of the two parts for each separate VIM section its open address and read or write control signals.

15. (original): The method of claim 13 further comprising the step of receiving and temporarily storing in said line buffer a plurality of data packets comprising a load/modify VLIW memory address (LV2) instruction and a plurality of short instruction words (SIWs) constituting a specified functional VIM portion to be loaded at an address specified in the LV2 instruction.

16. (canceled)

17. (new): A very long instruction word (VLIW) memory (VIM) direct memory access (DMA) apparatus comprising:

a plurality of functional execution units, each functional execution unit performing a distinct operation;

a VLIW memory (VIM) having a plurality of VIM sections, each VIM section storing instructions corresponding to each functional execution unit, each VIM section associated with a functional execution unit, each stored instruction selectable from any VIM section for parallel execution with any other stored instruction associated with a different functional unit of the plurality functional execution units;

a DMA very long instruction word (VLIW) line buffer for receiving a data packet having a plurality of instructions to a VIM section; and

a VIM load controller for separately controlling the loading of each received instruction of the plurality of instructions into a separate VIM section.

18. (new): The apparatus of claim 17 wherein each of the plurality of VIM sections has two ports allowing simultaneous read and write accesses.

19. (new): The apparatus of claim 17 wherein said line buffer temporarily stores the data packet comprising a load/modify VLIW memory address (LV2) instruction and a plurality of short instruction words (SIWs), each short instruction word of the plurality of short instruction words loaded into a VIM section at an address specified in the LV2 instruction.

20. (new): A method for loading very long instruction word (VLIW) memory (VIM) through a direct memory access (DMA) controller, said method comprising:

providing a VLIW memory (VIM) having a plurality of VIM sections, each VIM section storing instructions corresponding to a functional execution unit of a plurality of functional execution units, each stored instruction selectable from any VIM section for parallel execution with any other stored instruction associated with a different functional unit;

receiving a data packet having a plurality of instructions into a line buffer; and

selectively loading each instruction of the plurality of instructions from said line buffer to each corresponding VIM section of the plurality of VIM sectionsby utilizing a VIM load controller.

21. (new): The method of claim 20 wherein each separate VIM section has two ports allowing simultaneous read and write accesses, and the method further comprises:

separately providing each of the two parts for each separate VIM section its open address and read or write control signals.

22. (new): The method of claim 20 further comprising:

temporarily storing in said line buffer a plurality of data packets comprising a load/modify VLIW memory address (LV2) instruction and a plurality of short instruction words (SIWs), wherein the selectively loading step loads each VIM section sequentially, for each data packet, wherein the starting address of each VIM section is determined by the addresses contained in the LV2 instruction.